



# DRAG'N'DROP

RISC OS **Pi** and all RISC OS 5 machines

October 2015

Volume 7 Issue 1

£3.50

## Father Xmas's Crackers

Type-in desktop game!

## 3D Graphics

BBC Micro and RISC OS

Desktop retro gaming

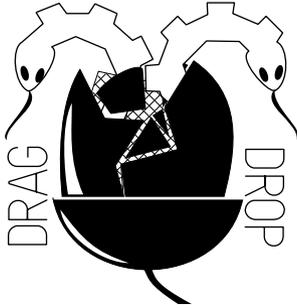
WFM2Draw



## Programming

Python Primary School C-Side

**The world's only regular RISC OS mag!**



Copyright © Drag 'N Drop 2015  
Produced on RISC OS computers

This issue has been blessed with contributions from the following people:  
Paul Dunnington (Python Primary School)  
Christopher Dewhurst (everything else)  
3D Graphics based on an article on *Home Computer Advanced Course*

The views expressed in this magazine are not necessarily those of the editor.

Alternative views are always welcome and can be expressed by either writing an article or a short editorial.

All articles and advertisements are published in good faith. No materials in this publication are meant to be offensive or misleading. If you come across something you believe is either of the above please contact the editor using the details below.

Contact Information  
Editor: Christopher Dewhurst  
Email: [editor@dragdrop.co.uk](mailto:editor@dragdrop.co.uk)  
[www.dragdrop.co.uk](http://www.dragdrop.co.uk)

# EDITORIAL

Hope you have been looking forward to Volume 7 of *Drag 'N Drop*.

We have the usual action packed mix of RISC OS news, serious articles including programming in Python and C and a bit of early festive cheer in the form of Father Xmas's Crackers (you know how early in the year he has to start get the stuff in the shops these days).

Special thanks to the writers especially if you are busy putting something together for the next issue!

Chris.

Christopher Dewhurst

QUO VADIS?



## At a glance...

Editorial	1
How do I...?	2
News & Apps	3
Files of the World	4
Python Primary School	10
Desktop Retro Gaming	18
Weblog	26
C-Side	27
3D Graphics	29
&CAFE	40
Father Xmas's Crackers	42

# How do I...?

## ...get the BBC Basic prompt?



To get the BBC Basic prompt press F12 and type \*BASIC and press Return. You can change the screen mode with MODE n where n is a number e.g. MODE 7 or MODE 0. Type AUTO for automatic line numbering. Press Escape to stop and type *SAVE "myprog"* followed by Return to store *myprog* on hard disc.

To return to the desktop type \*QUIT. Programs listed in *Drag 'N Drop* are assumed to work on all machines with RISC OS 5 e.g. Raspberry Pi, unless otherwise stated.

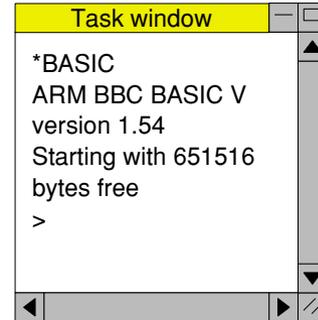
## ...open a Task window?

Menu click over the Raspberry icon on the right side of the iconbar and select click on Task window. Or press Ctrl + F12.



You may need to reserve more memory for the task in which case adjust-click on the Raspberry icon and under Application tasks click and drag the Next slide bar out to the right.

You can also type programs in a task window, hold down Ctrl and press F12. You can't use the cursor editing facility or change MODE, however.



You can also program and run Basic programs from the desktop. Double-clicking on the filer icon runs it, holding down Shift and double clicking loads it into your text editor.



## ...select the currently selected directory?

Articles may tell you to set the CSD (currently selected directory). Just click menu over filer window and choose Set directory ^W or you can use the IEasyCSD application presented in *Drag N Drop* 6i1.

## ...open an Application Directory?

Application directories begin with a ! called 'pling'. Hold down shift and double click select to open the directory

# News and Applications

## London 2015 Show

Saturday 24th October in Feltham is the date for the show in the south east of England, tickets just £5.00 on the door. You can always look forward to something exciting being released. Copies of Drag N Drop 7i1 will be available for the special price of £3.00

## Legends of Magic

A second offering from that vertiable master of RISC OS gameswriting, Tony Bartram, will be released at the London show. If his *Overlord* is anything to go by then *Legends* (a 3-D isometric adventure) will be worth the wait and definitely the cash.



## Quarterly 'disc'

The 'type in' programs in *Drag 'N Drop* have been available on the iStore version of the magazine for a while now. Some of you have been asking if you can buy the programs from our website too so we'll shortly be uploading them. Price likely to be an extra £1 to £1.50 for an archive (.zip) file for each issues from volume 6 on.

## ARMX6 Updates

Service pack 4 has been released for ArmX6 owners, the computer developed by R-Comp. Log in to the website with your registration details to obtain the latest updates.

## ROOL books

Three updated reference books, the Desktop Tools, Acorn C/C++, and Assembler (up to ARM v8) are available from ROOL priced £54 +P&P. More copies of the 2015 Style guide have also been printed and can be bought for £16

+P&P each.



## Bonus Binaries and a new User Guide

A suite of applications, modules and utilities have been added to ROOL site including window template and font editors (we'll be covering those in a future edition of *Drag 'N Drop*). Download free from [www.riscosopen.org.uk/content/downloads/common](http://www.riscosopen.org.uk/content/downloads/common) where a new edition of the RISC

OS 5 *User Guide* is also being drafted.

### **New RISC OS browser**

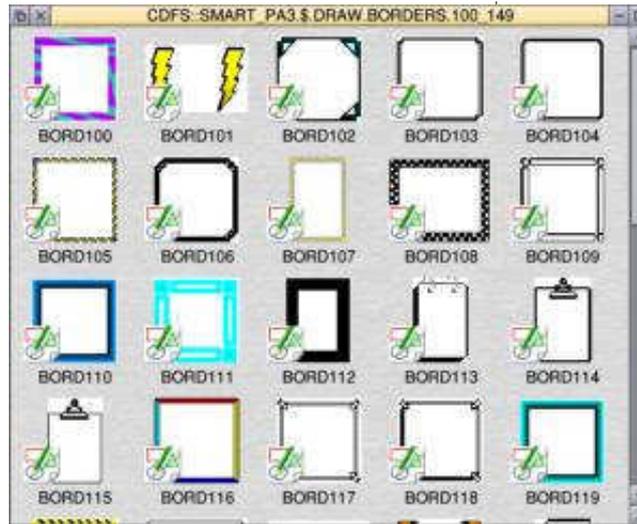
Otter is a new web browser being developed for RISC OS, test versions only available at [www.riscosports.co.uk/otter-browser\\_0.9.8-1.zip](http://www.riscosports.co.uk/otter-browser_0.9.8-1.zip). including several hundred megs of support code so Otter is currently recommended for only the fastest RISC OS machines.

### **GCC C**

Version 4.7.4 of the free C compiler is available from [www.riscos.info/index.php/GCC](http://www.riscos.info/index.php/GCC). There's never been a better time to start learning with C-Side, our series on programming in the language!

### **Thump**

Version 1.53 of the image viewer is available from [www.users.on.net/~belles/software/thump/](http://www.users.on.net/~belles/software/thump/). Drag a file of images to the iconbar and out will pop a window of thumbnails of images. A wide variety of formats are supported.



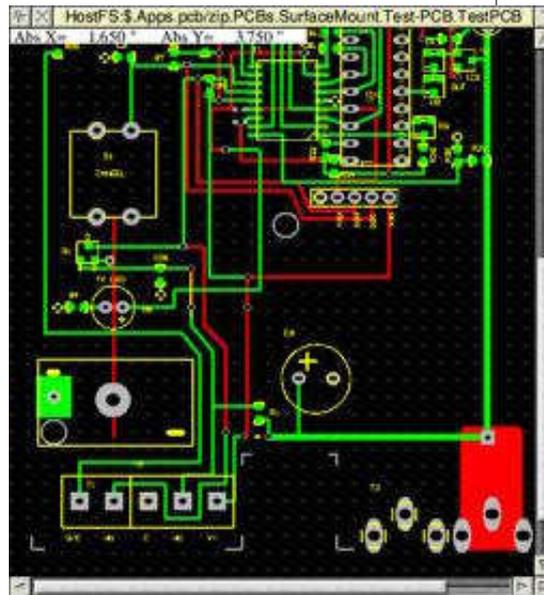
### **RiscPCB**

The popular Printed Circuit Board designer for RISC OS has been made Pi compatible and version 1.02 is available from [www.mbelect.co.uk/RISCOS](http://www.mbelect.co.uk/RISCOS).

### **AMPlayer for ARM7**

Some 'under-the-bonnet' changes

have been made to the modules for AMPlayer which means you can play MP3 files on your Pi without having to switch 'alignment exceptions off'. A fully 32-bit neutral version of AMPlayer can be downloaded for free at [www.riscos.info/index.php/AMPlayer](http://www.riscos.info/index.php/AMPlayer).



# Files of the World

## 3 Windows Metafiles

In this instalment of Files of the World we'll look at Windows Meta Files (WMF). WMFs are vector graphics like cartoons used in Powerpoint and Word. They are the Microsoft equivalent of Draw on RISC OS.

I'll present a listing for a WMF to Draw converter application. You then have access to literally thousands of WMFs available for free or little cost on the internet. I purchased the 'Clipart Multipack' of 10,000 WMFs for 99p including postage from [www.uk-surplus.com](http://www.uk-surplus.com).)



WMFs have a two headers 40 bytes large in total – a 20-byte

"Aldus placeable WMF" plus 20-byte Microsoft header.

Aldus was an American corporation which introduced the additional header and has since become more-or-less standardised.

The header begins with a 'magic number' of &9AC6CDD7 identifying it as a WMF, rather like Draw begins with &77617244 (the Ascii characters for 'Draw').

The only useful piece of information in the headers is at offset &0E. It gives the number of Metafile units per inch, for example 1440.

Coordinates are quoted in these Metafile units. We have to divide the co-ordinate by the Metafile units per inch and multiply by Draw units per inch which is always 46,080.

If you followed the 'Anatomy of Draw' series in *Drag 'N Drop* a couple of years back you will know that Draw uses four bytes (32 bits) to store coordinates of points in shapes.

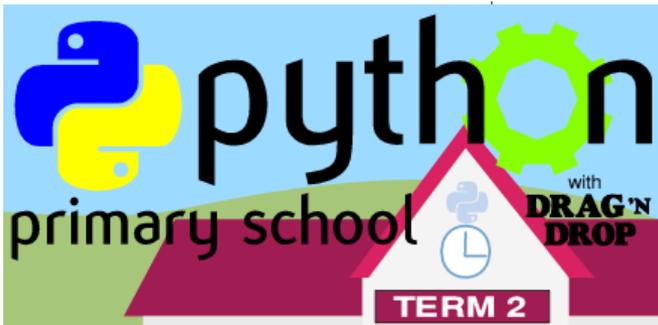
The 46,080 Draw units per inch are a bit of an overkill. Only two bytes (16 bits) or "half words" are needed to store coordinates in WMFs so they are more economic with space.

When converted to Draw, WMFs appear bloated because double the amount of space is used for coordinates – four bytes instead of two.

Beware WMF literature on the internet where the term "word" is used to describe two bytes of memory. On RISC OS a "word" has come to mean four bytes. I use the term 'half-word' to avoid any ambiguity.

After the header comes the 'body' of objects. Each object starts with a full word describing the number of half-words the object occupies (including the full word), a half-word specifying the object type, then a number of parameters – the exact number depending on the object type.

The Windows Metafile Format Specification can be found on the



I hope you've all done your homework of 100 lines? Click menu over Edit on the iconbar, select Create > python and press return. In the blank document type:

```
print 'I must not throw teddy in class'
```

Open *!Python.Lib*, inside you should have a subdirectory called *1Mine* (create it now if not). In Edit press F3, Ctrl+U (to get rid of "EditFile") and type *100Lines*, then drag the icon to the *1Mine* window to save it. Double click *100Lines* to run.

Well that's one line, how do we get a hundred? Find *Python-2/4-additions-2005-09-30/zip*, open it and drag *!PythonW* icon into your *Programming.Python* directory.

Double click it and the Python

Interpreter will open. In the box type **help()** and press Return. Then **keywords** and return.

Incidentally when I tell you to type commands assume you are to press the Return key after every command unless I tell you otherwise.

On line 4 you see mention of **while**. Type **while** to find out. Have you got that? Me neither! Let's add line 2 to 5 to our *100Lines* program. Add the following lines to *100Lines*

```
x = 1
while x < 2 :
    print x
    x = x + 1
```

Don't forget the colon, and the spaces which precede (indent) the last two lines. If you'll get an "IndentationError:" or unexpected results. As long as the indentation is consistent it doesn't matter to Python how many spaces we use.

Save *100Lines* and run it. Change the 2 to 4, save and try

again. We're on our way now!

Back in *!PythonW*, we have ignored the **while\_stmt ::=** mentioned in the keyword help bit and used the **while expression[1] ':' suite[2]**. The **else** isn't needed here but note the first e' of **else** would need to be aligned with the w of **while** – them 'dents again.

Modify *100Lines* as follows:

```
line = 'I must not throw teddy in class'
x = 1
while x < 101 :
    print line
    x = x + 1
```

Save that and run – yep, 1 to 100.

How can we our homework on paper though? In *PythonW* type **print** for information on the print statement (you need to scroll up to see the beginning).

In the third paragraph it says **print** outputs by default to **sys.stdout** unless we use the extended form.

Type **sys.stdout** in *PythonW*. Now you've gone and done it! *PythonW* has a limited scroll range so we'll call up the same

# Desktop Retro Gaming

Thanks to the consistency of BBC Basic it is possible to take a 30 year old listing written for the BBC Micro and run it, without modification, in single tasking mode on RISC OS on the Raspberry Pi.

Simple games can also be run in a desktop window with the help of !GraphTask, available free from [www.armclub.org.uk/free](http://www.armclub.org.uk/free). You can't use sprites though.

In this article I'll demonstrate how to go about recreating the retro experience, sprites and all, on the desktop.

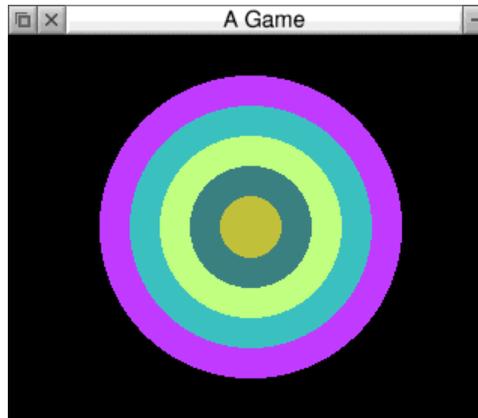
Some Wimp programming is involved by necessity but don't worry – it isn't as complicated as programming desktop applications.

We won't bother installing an icon on the iconbar, for example. The windows will have a close button and when that's clicked the whole game will quit.

The main thing we need to know is that we have to set the flag to tell the Wimp manager that

we'll do the work of updating the contents of the window ourselves. We also make use of OS\_SpriteOp SYSTEM calls to display sprites designed in one mode (e.g. Mode 5) in another mode, the mode your desktop runs in.

We'll discuss sprites shortly but first of all type in Listing 1 which is a quick multi-tasking demo.



## Listing 1

```
10REM Desktop Retro Gaming
20REM Program 1 demo window
30REM (c) Drag N Drop 2015
40
50ON ERROR PROCerror
60task$="A Game"
```

```
70width%=640:depth%=512
80SYS "Wimp_Initialise",500,&4B
534154,task$
90PROCwindow
100PROCscreen
110!block%=window%
120SYS "Wimp_GetWindowState",,block%
130SYS "Wimp_OpenWindow",,block%
140finished%=FALSE
150
160REPEAT
170SYS "Wimp_Poll",0,block% TO event%
180CASE event% OF
190WHEN 1:PROCredraw
200WHEN 2:SYS "Wimp_OpenWindow",,block%
210WHEN 3,17,18:finished%=TRUE
220ENDCASE
230UNTIL finished%
240SYS "Wimp_CloseDown"
250END
260:
270DEF PROCredraw
280SYS "Wimp_RedrawWindow",,block% TO more%
290WHILE more%
300PROCscreen
310SYS "Wimp_GetRectangle",,block% TO more%
320ENDWHILE
330ENDPROC
340:
350DEF PROCscreen
360PROCgetorigin
370GCOL 0,128
380RECTANGLE FILL ox%,oy%,width%
```

# Weblog: Gumtree

If you like bargain hunting on the internet but are frustrated with eBay now it's largely an outlet for commercial products then it could be that Gumtree is the alternative you're looking for.



It's based in the UK and claims to have over a million classifieds. It's at [www.gumtree.com](http://www.gumtree.com) and works on Netsurf on RISC OS. Well, it works okay *ish*.

I haven't tried selling anything yet but there are no fees and I've browsed for and bought a few appliances from the local community with no problem..

You simply text or ring the seller's number (shown clearly on the ad's page – no Captchas etc.) or use the messaging system to arrange collection. Gumtree has no 'feedback' mechanism which its critics argue is a weakness.

"Meet face to face, never send payments for items you've not seen" is the advice given on the

site and that's no different from the traditional paper classifieds mag you get through the door so it's buyer beware.

An account with Gumtree is required but only your name and email address is needed.

[www.gumtree.com/for-sale](http://www.gumtree.com/for-sale) is the place to start looking. Type a keyword to begin the search. You can click on *England* then *Suffolk* and *Ipswich* say to narrow down the search area to help decide how far you want to/can travel.



I've found that only parts of photos are visible but this can be overcome by menuing over the photo and choosing Object > Save and drag-dropping the

JPEG picture back onto Netsurf's iconbar icon.



To contact the seller you can click on the orange Email button.

Again once you've bought your item there's no feedback but as you will have exchanged contact details with the seller you can get in touch if there is a problem, however I've had no trouble with my Gumtree purchases. ■



We spent the last two sessions getting up and running with both C compilers available for RISC OS, the free GCC and Acorn C (sometimes called Norcroft C).

This time at the C Side we're jumping in at the deep end and going to write a program to convert arabic numerals to roman numerals.

You see roman numerals on some clock faces at the end of television credits to give the year e.g. 2015 is MMXV.

I'll give both the C and equivalent Basic version now. Line numbers in the C version are given for reference only, they shouldn't be typed in.

### roman.c listing

```

1// converts arabic to roman
2#include <stdio.h>
3int main(void)
4{
5    int i
6    v[ ] = {1000, 900, 500, 400,
100, 90, 50, 40, 10, 9, 5, 4, 1};
7    char *r[ ] = {"M", "CM", "D",
"CD", "C", "XC", "L", "XL", "X",
"IX", "V", "IV", "I"};
8    int n;
9    printf("Enter the Arabic
number: ");
10 scanf("%d", &n);
11 printf("In Roman, ");
12
13 for(i=0; i<=13; i++)
14 {
15     while(n >= v[i])
16     {
17         printf("%s", r[i]);
18         n -= v[i];
19     }
20 }
21 return(0);
22}

```

### ROMAN listing

```

10REM converts arabic to roman
20DIM v(13), r$(13)
30v( )={1000,900,500,400,100,90,5
0,40,10,9,5,4,1
40r$( )="M" "CM" "D" "CD" "C" "X
C" "L" "XL" "X" "IX" "V" "IV" "I"
50INPUT "Enter the Arabic numbe
r",n
60PRINT "In Roman, ";
70FOR i=0 TO 12
80WHILE n>=v(i)
90PRINT r$(i);

```

```

100n-=v(i)
110ENDWHILE
120NEXT
130

```

Save the C listing as *roman* (no .c suffix) in your *Programming.c* directory. Make *Programming.c* is the currently selected directory.

If you are using Acorn open the *AcornC/C++* folder so the computer has 'seen' it. If you are using GCC double click !GCC and ensure the Next slider in the task display is at least 4000k

Press Ctrl+F12 to open a task window. Type

```
cc roman.c -o object
```

for Acorn C or for GCC

```
gcc roman.c -o object
```

All being well you can now double click *object* and try the program:

```
Enter the Arabic number: 1987
In Roman, MCMLXXXVII
```

We already know about comments, the lines beginning with /, which are like Basic REM statements. We also met the *include* and *scanf* statements last time, The program in the last instalment used a FOR loop so you should be getting confident in their use. If you are at all unsure

# 3D Wireframe Graphics

**3D wireframe graphics BBC Basic are straightforward. In this series you can learn how to draw and manipulate three-dimensional images.**

Programs will work on all Acorn and RISC OS machines from BBC to RISC OS Pi. Please note, though, the older the machine the slower the drawing.

The normal method of drawing lines on the computer is similar to the way you use a pen and paper.

The main difference between the computer and a person using pen and paper is that the computer is faster and better at drawing straight lines.

You can draw outline images where objects are made from a wire frame joined to form a grid or mesh.

The basic steps are in a set of routines, which include the BBC Basic drawing commands – PLOT, MOVE and DRAW.

To set up the first stage of these drawing routines, enter this section of programming, but do

not RUN it yet, because it is incomplete.

```
9000DEF PROCinit
9010CLS:CLG
9030ENDPROC
9040:
9100DEF PROCmove(x,y)
9110MOVE x,y
9120ENDPROC
9130:
9200DEF PROCdraw(x,y)
9210DRAW x,y
9220ENDPROC
9230:
```

The simplest shape is a line so here is a routine to do it (you still won't be able to RUN the program):

```
9500DEF PROCline(xs,ys,xe,ye)
9510PROCmove(xs,ys)
9540PROCdraw(xe,ye)
9550ENDPROC
```

To depict a surface and any irregularities such as cracks, hills and dales it is best to visualize it as a grid of horizontal and vertical lines. Irregular features in the surface can be shown as distortions of these lines. Enter the following (but do not RUN) to define a PROCedure to draw a grid:

```
5000DEF PROCgrid(xa,ya,lw,lh,
nx,ny)
5010ja = lw/nx
5020xs = xa
5030FOR jb = 0 TO nx
5040PROCline(xs,ya,xs,ya + lh
)
5050xs = xs + ja
5060NEXT
5100ja=lh/ny
5110ys = ya + lh
5120FOR jb = 0 TO ny
5130PROCline(xa + lw,ys,xa,ys
)
5140ys = ys - ja
5150NEXT
5160ENDPROC
5200:
```

Coordinates  $(xa,yb)$  specify the bottom lefthand corner of the grid.  $lw$  is the width, and  $lh$  the height,  $nx$  the number of horizontal divisions, and  $ny$  the number of vertical divisions.

$ja$  specifies the distance between the vertical lines, and the FOR..NEXT loop draws horizontal lines stepped off in this distance. The second FOR..NEXT loop draws vertical lines in steps calculated by line 5100.

Users of BBC Basic on RISC OS machines can use the BBC

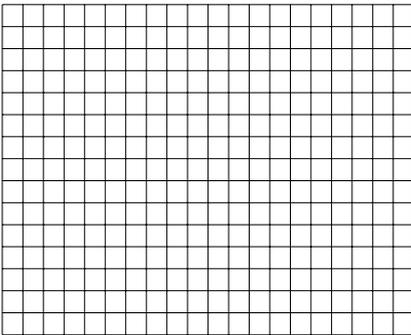
Basic V shorthand to abbreviate the line  $x_s = x_s + ja$  to  $x_s += ja$ .

PROCgrid can map only flat surfaces. It could not, for example, depict irregularities within the surface.

To display the grid on the screen, enter these lines to call the routine and RUN the program:

```
100MODE 4
110PROCinit
180PROCgrid(0,0,1279,1023,20
,15)
190END
200:
```

When you RUN the program, you should see a grid filling the entire screen.



To see how versatile the program is, make the changes below and RUN again:

```
180PROCgrid(0,0,1279,1023,1,
1)
```

A rectangular grid is not the only 'building block': it's often useful to be able to draw circles.

BBC Basic V has a CIRCLE command but does not give the control you need for drawing three-dimensional images. With perspective, a circle in one view might be an ellipse in another, or some other curve in others.

There is also an ELLIPSE command but it cannot cope with the third dimension, which is essential for natural-looking shapes. So it is better to be able to define a general function.

One way to draw a circle is with a series of short, straight-line sections. Provided the lines are short, the circumference of the circle will appear as a smooth curve.

Here is a routine to draw a circle radius  $r$ , with centre at  $(x_s, y_s)$ ; do not RUN the program yet:

```
6000DEF PROCcircle(xs,ys,r,n)
6010IF n = 0 THEN n = 20+ INT
(r/10)
6020ja = 2*PI/n
6050PROCmove(xs + r,ys)
6060jb = 0
6070FOR jc = 2 TO n
6080jb += ja
```

```
6110PROCdraw(xs + r*COSjb, ys
+ r*SINjb)
6120NEXT
6150PROCdraw(xs + r,ys)
6160ENDPROC
6170:
```

$n$  sets the number of straight-line segments to be used for the circumference of the circle. If you specify  $n = 0$ , line 6010 calculates how many segments are needed for the smoothest circle, taking into account the size of the display.

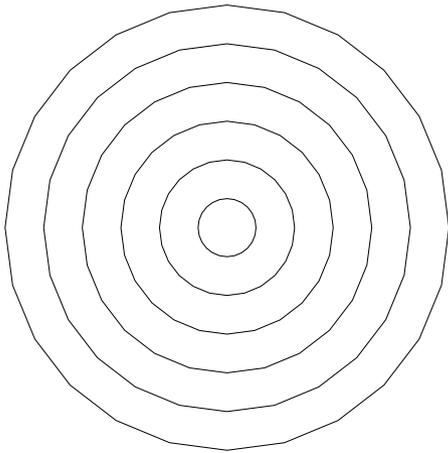
Line 6020 calculates the angle of each line segment on the circumference. Line 6050 moves the graphics cursor to a position on the circumference.

The FOR ... NEXT loop draws each line segment, except the last one, which is drawn by 6150 to ensure that the last line joins up with the first.

To see how PROCcircle works, delete line 180 and add the next few lines to call it:

```
150FOR r = 60 TO 500STEP 80
160PROCcircle(640,512,r,24)
170NEXT
```

Your screen should now show a number of concentric circles. As with PROCgrid, you can vary the



parameters in the program that calls it to change the display. Line 150 sets the radius of the first circle, and the amount by which it is increased to give the radii of successive circles. Line 160 specifies the centre of the circles and the number of line sections in the circumference.

As an exercise, vary these values and note the effect on the display.

If you are wondering what has happened to PROCgrid, it is still in the memory, but since you have rewritten the lines of code that call it, the computer does not display it. PROCedures like these can be collected into a library of useful line graphics routines for

use as needed. They can be used in all sorts of graphics programs, as well as those for wireframe drawing.

In theory, at least, it is nonsense to say that you can create a three-dimensional object (a box, say) on a two-dimensional medium— whether this is a piece of paper or a TV screen.

But what you can do is to employ a visual convention so that our eyes perceive or 'read' a series of lines as a representation of a solid object.

A painting of a landscape may appear to have depth and distance because the artist has employed visual trickery – perspective.

Perspective depends on the fact that we understand objects to get smaller as they move into the distance, and that parallel lines appear to converge as they get farther from the viewer.

Perspective isn't the only convention – in technical drawing, it is common to use a convention called *isometric projection*. Lines which recede from the viewer are drawn at an angle (as in

perspective drawing).

Unlike perspective drawing, however, these lines do not converge, nor do objects get smaller as they get farther away. This has a number of advantages for technical applications. You can take direct measurements of any line because its scale isn't affected by its direction.

Although you can represent perspective on the computer display this is harder to set up than an isometric projection. Isometric drawing relies simply on setting up a third axis on the screen.

You already have an *X* axis (horizontal) and a *Y* axis (vertical). All you need now is a third axis, the *Z* axis, which is at an angle to the first two. Any line which is drawn at this angle is then understood to represent a line moving away from (or towards) the viewer. This is much simpler to understand if you look at figure 1.

One way to produce 3D wireframe images is to build them up from 2-D shapes, using PROCgrid and PROCcircle.

# &CAFE

Welcome to the &CAFE, you say it "Ampersand Cafe". Why not come in for a coffee and chat about all things hexadecimal?

You'll have heard of computer geeks talking about hexadecimal or hex for short. What's all this hex malarky? To understand we first have to get our heads round binary numbers.

You'll probably hear the geeks say computers work in binary numbers, 1s and 0s. Now, we human beings work in decimal numbers. Binary and hexadecimal numbers are just another way of counting the same thing, except they are more convenient for machines.

In our decimal system we have ten digits or symbols – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Above nine we string together two or more digits to represent bigger numbers, e.g. 19 (nineteen) is 1 lot of 10 add 9.

In binary there are only two digits: one and zero. For numbers above one the computer strings together two or more digits.



0 binary is 0 decimal, 1 binary is 1 decimal but then we have to start combining digits so 10 is decimal 2, 11 is decimal 3 and so on.

In BBC Basic we use the percent symbol (%) to indicate binary. So %11 is decimal 3, %100 is decimal 4, %101 is decimal 5 and so on. We can carry on counting this way ad infinitum, but let's stop at %1111 binary which is decimal 15.

Notice it's a group of four digits. Or as the geeks would say, four *bits*. This is where hexadecimal comes in. All 16 (decimal) numbers can be

represented by just one hexadecimal symbol. The first 10 symbols in hexadecimal are exactly the same as decimal symbols, namely 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. But then we use capital letters: A,B,C,D,E,F. That's 16 symbols in all for 16 values.

The table below shows the first 16 numbers in decimal, binary and hexadecimal.

Decimal	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

# Father Xmas's Crackers

Father Christmas is out delivering presents when he discovers the mischievous gnomes have rigged the crackers with explosives! Can you defuse them all in time?



Move around with the cursor keys. Press shift in conjunction with the keys to move left or right the row of blocks which Father Xmas is on except when there is a cracker on the row. If you fail to defuse the cracker before its timer runs out you lose a life. The game ends when you've lost all your

lives.

Beware the deadly holly and mutant stockings which roam the screens on later levels. But drink the brandy for bonus points.

To type in the listing create an application directory (!Xmas). Type in listing 1 and save it inside !Xmas. Ensuring !Xmas is the currently selected directory double click to run it. Then create a short !Run file which outputs the sprites as XMAS1.

Create a short Obey file with the following lines:

```
!Run file for Father Xmas's  
Crackers  
!(c) Drag N Drop 2015  
Set xmas$dir <obey$dir>  
Run <xmas$dir>.XMAS2
```

Now type in listing 2 ensuring there are no errors and save it as XMAS2. Double click !Xmas to

run.

If you have User Voices module installed (see Drag 'N Drop 5i3) you will get the sound effects.

## Listing 1

```
10REM Father Christmas's Cracke  
rs  
20REM RISC OS version  
30REM Generate Sprites  
40REM (c) Drag N Drop 2015  
50  
60DIM S% &1000,C% 3  
70N%=13  
80!S%=N%:S%!4=&10  
90I%=12  
100FOR J%=0 TO N%-1  
110READ H%,V%,!C%:Z%=H%*4+V%  
120I%!(S%+&2C+Z%  
130$(S%+I%+4)=STR$J%+STRING$(12-  
LENSTR$J%,CHR$0)  
140I%!(S%+16)=H%*4 DIV4-1  
150I%!(S%+20)=V%-1  
160I%!(S%+24)=0:I%!(S%+28)=&1F  
170I%!(S%+32)=&2C:I%!(S%+36)=&2C  
180I%!(S%+40)=13  
190X%=0:FOR L%=1 TO V%:READ A$  
200IF A$="0" A$=STRING$(H%*4,"0"  
)  
210FOR B%=1 TO LEN A$  
220D%=VALMID$(A$,B%,1)  
230X%?(S%+I%+&2C)=D%?C%  
240X%=X%+1:NEXT NEXT  
250I%=I%+Z%+&2C:NEXT
```

```

2340FOR a%=0 TO 1%:PROCudg(156,x%
,y%+a%,0):PROCudg(158,x%+w%,y%+a%,
0):NEXT
2350FOR a%=0 TO w%:PROCudg(156,x%
+a%,y%,0):PROCudg(157,x%+a%,y%+1%,
0):NEXT
2360PROCudg(155,x%+w%,y%,0):PROCU
dg(155,x%,y%+1%,0):PROCudg(154,x%+
w%,y%+1%,0)
2370ENDPROC
2380DEF FNz(A%,X%)=LEFT$(STRING$(
A%,"0"),A%-LENSTR$X%)+STR$X%
2390DEF PROCreset
2400K%=0:V%=3:S%=0:D%=FALSE:T1%=T
IME:T2%=9000
2410ENDPROC
2420DEF PROCinit
2430VDU23,159,0,0,144,217,255,255
,231,255
2440DIM map% 196,xs%(3),ys%(3),fs
%(3),dx%(3),dy%(3),ex% 7
2450FOR X%=0 TO 4:READ X%?ex%:NEX
T
2460dx%()=-1,1,0,0:dy%()=0,0,-1,1
2470DIM trtab% &100:SYS "ColourTr
ans_SelectTable",13,0,-1,-1,trtab%
2480scale%=2
2490DIM sctab% 16:!sctab%=scale%:
sctab%!4=scale%: sctab%!8=1: sctab%!
12=1
2500size%=4000
2510DIM area% size%
2520!area%=size%:area%!8=16:SYS "
OS_SpriteOp",&109,area%
2530SYS "OS_SpriteOp",&10A,area%,
"<xmas$dir>.XMAS1"
2540chc%=32:VOICES 7
2550FOR X%=1 TO 7:OSCLI"CHANNELVD
ICE "+STR$X%+" "+STR$(X%+9):NEXT
2560ENDPROC
2570DEF PROCwindow
2580DIM block% &100
2590P%=block%
2600width%=1280:depth%=960
2610LOPT 2
2620EQUID 100
2630EQUID 100
2640EQUID 100+width%
2650EQUID 100+depth% ;visible area
2660EQUID 0
2670EQUID 0
2680EQUID -1
2690EQUID &87001002 ;flags
2700EQUID &01070207
2710EQUID &00000103 ;window colour
2720EQUID 0
2730EQUID -depth%
2740EQUID width%
2750EQUID 0 ; work area extent
2760EQUID &19
2770EQUID &F000 ;title bar and wor
k area flags
2780EQUID 0
2790EQUID 0 ;sprite area pointer
2800EQUIS game$+CHR$13
2810EQUID 0 ;no icons
2820ALIGN
2830J
2840SYS "Wimp_CreateWindow",,bloc
k% TO window%
2850ENDPROC
2860DEF PROCerror
2870!block%=ERR
2880$(block%+4)=REPORT$+" at line
"+STR$ERL+CHR$0
2890SYS "Wimp_ReportError",block%
,1,game$
2900SYS "Wimp_CloseDown"
2910END
2920DATA 0,7,8,7,0

```

Have YOU written a program for your RISC OS Pi?



Can YOU write an article to describe it?

Get £15 to appear in Drag 'N Drop!

dragdrop@dragdrop.co.uk

**LUTIUS**  
project

The complete BBC Micro works of Chris Dowhurst 1995-2005

**10th Anniversary CD-Rom**

£1 p+p anywhere in world



Buy from website at dragdrop@dragdrop.co.uk